

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Grafické uživatelské rozhraní pro databázový systém  
RESIQL



2015

Vedoucí práce:  
Mgr. Petr Krajča, Ph.D.

Martin Moško

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Martin Moško  
Název práce: Grafické uživatelské rozhraní pro databázový systém RESIQL  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2015  
Studijní obor: Aplikovaná informatika, prezenční forma  
Vedoucí práce: Mgr. Petr Krajča, Ph.D.  
Počet stran: 51  
Přílohy: 1 CD/DVD  
Jazyk práce: slovenský

## **Bibliographic info**

Author: Martin Moško  
Title: Graphical user interface for database system RESIQL  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2015  
Study field: Applied Computer Science, full-time form  
Supervisor: Mgr. Petr Krajča, Ph.D.  
Page count: 51  
Supplements: 1 CD/DVD  
Thesis language: slovak

## Anotácia

*Bakalárska práca bola zameraná na vylepšenie a zjednodušenie práce s RESIQL databázami. Výsledkom je nová aplikácia určená na ich správu. Užívateľ nemusí podrobne poznať syntax databázového jazyka, aby bol schopný vytvárať alebo spravovať relačné tabuľky databázy.*

## Synopsis

*Bachelor thesis was focused on improving and simplifying work with RESIQL databases. The result is a new application for database administration. User does not need to know syntax details of a database language to be able to create or manage database relational tables.*

**Kľúčové slová:** Java; RESIQL; Počítačová aplikácia; Užívateľská a technická príručka

**Keywords:** Java; RESIQL; Desktop application; User and technical documentation

Ďakujem vedúcemu práce za skvelú spoluprácu a rady. Svojej rodine a priateľke za psychickú podporu a trpezlivosť pri tvorbe tejto práce.

*Čestne vyhlasujem, že som celú prácu vrátane príloh vypracoval/a samostatne a za použitia iba zdrojov spomínaných v texte práce a uvedených v zozname literatúry.*

dátum odovzdania práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Jazyk RESIQL</b>	<b>10</b>
<b>3</b>	<b>Užívateľská príručka</b>	<b>11</b>
3.1	Kompatibilita . . . . .	11
3.2	Inštalácia a prvotné spustenie . . . . .	12
3.3	Pripojenie RESIQL Manager aplikácie k databáze . . . . .	13
3.4	Popis hlavného okna . . . . .	14
3.5	Stromová štruktúra databázy . . . . .	15
3.5.1	Tabulky . . . . .	16
3.5.2	Pohľady . . . . .	18
3.5.3	Domény . . . . .	19
3.5.4	Operátory . . . . .	20
3.6	Editor dotazov . . . . .	22
3.7	Tabulkový editor . . . . .	24
3.7.1	Režim detailov . . . . .	24
3.7.2	Záložka konzoly . . . . .	25
3.7.3	Záložka tabulky . . . . .	26
3.7.4	Režim tabulkovej plochy . . . . .	27
3.7.5	Správa tabulkových dát . . . . .	27
3.7.6	Filtrovanie dát . . . . .	29
3.7.7	Návrh tabuliek . . . . .	31
3.7.8	Tvorba relácii tabuliek . . . . .	32
3.7.9	Ostatné funkcie tabulkového editora . . . . .	33
3.8	Nastavenia aplikácie . . . . .	35
3.9	Prídavné moduly . . . . .	36
<b>4</b>	<b>Technická príručka</b>	<b>37</b>
4.1	Vývojárske nástroje a materiály . . . . .	37
4.2	Aplikácia RESIQL Manager . . . . .	38
4.2.1	Štruktúra projektu ResiqlManager . . . . .	38
4.2.2	Inicializačný proces aplikácie . . . . .	39
4.2.3	Hierarchia základných ovládačov . . . . .	40
4.3	Knižnica ModuleLibrary . . . . .	41
4.3.1	Štruktúra projektu ModuleLibrary . . . . .	41
4.4	Použitie vybraných komponent a tried . . . . .	41
4.4.1	Práca s databázou . . . . .	42
4.4.2	Použitie ProgressWorker komponenty . . . . .	43
4.5	Vytvorenie prídavného modulu . . . . .	44
	<b>Záver</b>	<b>47</b>

Conclusions	48
A Obsah priloženého CD	49
Zoznam skratiek	50
Literatúra	51

## Zoznam obrázkov

1	Hlavné okno aplikácie . . . . .	11
2	Rozdelenie panelov hlavného okna . . . . .	14
3	Grafické zobrazenie aktuálneho stavu databázy . . . . .	15
4	Editor novej tabuľky . . . . .	16
5	Editor stĺpca tabuľky . . . . .	17
6	Náhľad tabuľky . . . . .	17
7	Editor pohľadu . . . . .	18
8	Editor domény . . . . .	19
9	Nápoveda domény . . . . .	19
10	Editor operátora . . . . .	20
11	Nápoveda operátora . . . . .	21
12	Editor dotazov . . . . .	22
13	Editor dotazov s chybou . . . . .	23
14	História dotazov . . . . .	23
15	Tabuľkový editor (Režim detailov) . . . . .	24
16	Tabuľkový editor (Režim tabuľkovej plochy) . . . . .	24
17	Záložka konzoly . . . . .	26
18	Reprezentácia tabuľky (tabuľková plocha) . . . . .	27
19	Správa tabuľkových dát . . . . .	28
20	Grafické označenie riadkov tabuľky . . . . .	29
21	X najlepších riadkov . . . . .	29
22	Riadky nad ohodnotením . . . . .	29
23	Radenie podľa . . . . .	30
24	Jednoduchá podmienka . . . . .	30
25	Vlastná podmienka . . . . .	31
26	Pridanie stĺpca . . . . .	31
27	Premenovanie stĺpca . . . . .	32
28	Pridanie predpony . . . . .	32
29	Vlastný operátor . . . . .	33
30	História zmien tabuľky . . . . .	34
31	Okno nastavení aplikácie . . . . .	35
32	Umiestnenie prídavných modulov . . . . .	36
33	Hierarchia základných ovládačov . . . . .	40

## Zoznam tabuliek

1	ASCII – UNICODE operátory . . . . .	22
---	-------------------------------------	----

## Zoznam zdrojových kódov

1	Vykonanie dotazu nad databázou . . . . .	42
---	--	----

2	Použitie ProgressWorker komponenty . . . . .	43
3	Vytvorená trieda Main.java . . . . .	44
4	Zapísanie textu do konzoly . . . . .	45



# 1 Úvod

Motiváciou na vznik práce bolo vytvorenie užívateľsky prívetivého nástroja na prácu s RESIQL databázami. Má poskytovať pohodlnú správu databáz pre nováčikov, ktorí nemajú skúsenosti so syntaxou a sémantikou [jazyka RESIQL](#), ako aj pre pokročilých užívateľov.

Vytvoril som aplikáciu, ktorá v aktuálnej verzii pokrýva základnú funkcionality jazyka v rozsahu možností databázového systému RESIQL. Systém je vyvíjaný Katedrou informatiky na Prírodovedeckej fakulte Univerzity Palackého v Olomouci a hlavným správcom je Mgr. Petr Krajča, Ph.D.

Dokument slúži ako užívateľská a technická príručka pre užívateľov aplikácie na správu RESIQL databáz.

Na začiatku v krátkosti predstavím jazyk RESIQL a jeho prínos.

Práca je následne rozdelená do dvoch častí: Užívateľská a Technická príručka. Najprv vás oboznámim s jednotlivými vlastnosťami a funkciami aplikácie z užívateľského pohľadu. Tato príručka čitateľovi poskytne dostatočné znalosti, aby mohol začať aplikáciu plnohodnotne používať.

Druhá časť je určená pre programátorov, ktorí by sa chceli podieľať na ďalšom vývoji aplikácie. Predstavím štruktúru zdrojových kódov projektu. Popíšem jednotlivé časti jadra aplikácie a vysvetlím, ako pracovať s rozhraním pre prídavné moduly.

## 2 Jazyk RESIQL

Pre lepšie pochopenie významu aplikácie je vhodné predstaviť jazyk RESIQL<sup>1</sup>. Je to relačný dotazovací jazyk, ktorý podporuje podobnostné dotazy a približné zhody. Formálny model jazyka je popísaný v publikáciach [1, 2]. Navrhnutý jazyk implementuje časť relačných operácií ako rozšírenie Coddového relačného modelu dát [3] a dovoľuje formulovať a vykonávať dotazy typu „Ukáž všetky záznamy, kde hodnota vlastnosti  $y$  je podobná hodnote  $d$ “.

Ako príklad si predstavte databázu ojazdených áut s vlastnosťami ako „typ auta“ a „cena auta“. Následne by ste chceli vyhľadať autá, typu SUV (alebo im podobné) s cenou nižšou ako 4000 eur (alebo trochu vyššou). V tom prípade by bolo vhodné mať jazyk, ktorý by bol schopný vyjadriť výraz či podmienku spôsobom ako:

type  $\approx$  SUV AND (price < 4000 OR price  $\approx$  4000)

Typ (type) a cena (price) sú vlastnosťami a  $\approx$  je operátor interpretovaný ako „byť podobný“.

Tradičné relačné databázové jazyky dokážu vyjadriť len dotazy založené na rovnosti, buď zhodu nájdu alebo nie. V RESIQL modeli má každý záznam (n-tica tabulkových dát) priradené ohodnotenie (rank) vyjadrujúce do akej miery sa zhoduje s dotazom založeným na podobnosti. Po vykonaní dotazu môžete skontrolovať tie najlepšie zhody a nájsť odpoveď, ktorá najviac vyhovuje vašim potrebám. Na to aby bolo možné takéto dotazy vykonať v praxi je nutné implementovať dotazovací jazyk s podporou podobnostne založených dotazov.

Jednou zo zvažovaných ciest bolo vytvoriť vrstvu nad tradičnou databázou [4], čo ale malo niekoľko nevýhod:

1. Výsledný jazyk je zvyčajne ťažkopádny a limitovaný základným jazykom
2. Je prakticky nemožné poskytnúť optimalizácie dotazu, pretože základný DBMS<sup>2</sup> je optimalizovaný na bežné dotazy založené na rovnosti, ktoré nemôžu poskytnúť funkcie rozšíreného modelu.

Z týchto dôvodov bol navrhnutý nový jazyk, ktorý kombinuje vlastnosti SQL a Tutorial D [5] a podporuje podobnostné založené dotazy a ohodnotenie (rank) od samého začiatku. [6]

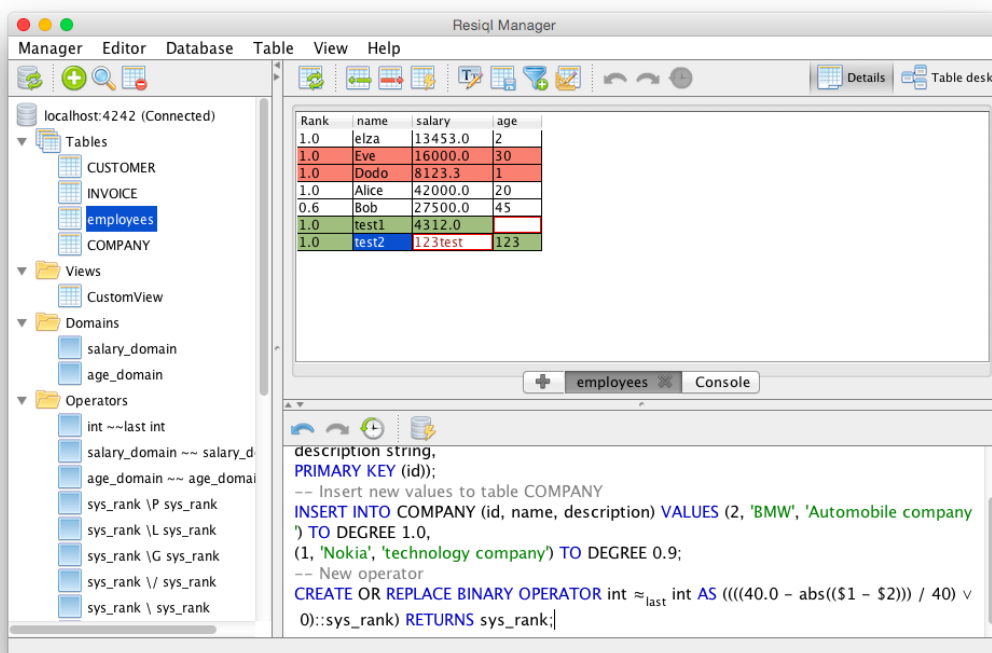
---

<sup>1</sup>Relational Similarity-based Query Language (RESIQL)

<sup>2</sup>Database management system (DBMS)

## 3 Uživatelská príručka

Príručka v nasledujúcich kapitolách popisuje vlastnosti a použitie aplikácie *RESIQL Manager* z pohľadu užívateľa. Úvod je venovaný príprave potrebných programov a inštalácii aplikácie. Následne budú predstavené jednotlivé časti aplikácie, druh informácií a funkcie, ktoré ponúkajú. Záver užívateľskej príručky sa venuje prídavným modulom, ich správnym nastavením a používaním.



Obr. 1: Hlavné okno aplikácie

### 3.1 Kompatibilita

Aplikácia je napísaná v jazyku Java<sup>3</sup>. To zaručuje kompatibilitu s každým operačným systémom, kde je podporované behové prostredie javy (JRE<sup>4</sup>). Pre správne fungovanie aplikácie je potrebné mať nainštalovanú minimálne verziu 8 a novšiu. Aktuálnu verziu inštalátora behového prostredia je dostupná na oficiálnych stránkach Javy: <http://java.com/en/download/>.

<sup>3</sup>[http://en.wikipedia.org/wiki/Java\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/Java_%28programming_language%29)

<sup>4</sup>Java Runtime Enviroment (JRE)

## 3.2 Inštalácia a prvotné spustenie

Inštalácia aplikácie je jednoduchá. Jedinou požiadavkou je nainštalovaná správna verzia behového prostredia Javy (viď. 3.1).

Aplikácia neobsahuje zabudovaný databázový RESIQL server. Ak prístupujete k vzdialenému serveru, nie je potrebné nič nastavovať. V prípade, že si situácia vyžaduje prácu na lokálnom databázovom serveri, je nutné server spustiť nezávisle na *RESIQL Manager* aplikácii.

Nasledujúci návod popisuje inštaláciu a spustenie aplikácie spolu s lokálnym databázovým serverom:

1. Z priloženého CD k bakalárskej práci si skopírujte zip archív zo zložky *bin/ResiqlManager-1.0.0-db.zip*
2. Archív rozbalte na mieste odkiaľ budete chcieť aplikáciu a databázový server spúšťať.
3. Otvorte príkazový riadok alebo terminál nad zložkou, kde sa nachádza serverová aplikácia **resiql-0.3.3.2.jar**
4. Do príkazového riadku zadajte nasledujúci príkaz na spustenie databázového servera:

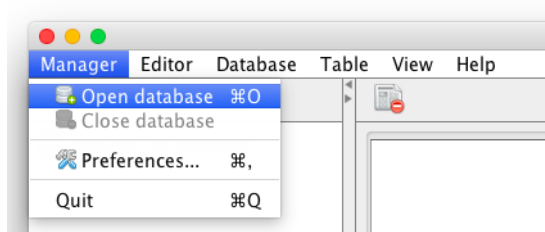
```
java -jar resiql-0.3.3.2.jar -s
```

5. Ak úspešne prebehlo spustenie servera, môžete spustiť *RESIQL Manager* aplikáciu **ResiqlManager-#.#.#.jar**

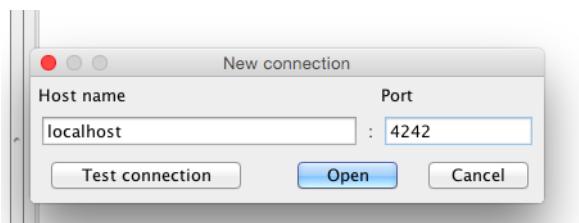
### 3.3 Pripojenie RESIQL Manager aplikácie k databáze

Existuje viac spôsobov ako otvoriť databázu v aplikácii. Uvedený je jeden z nich:

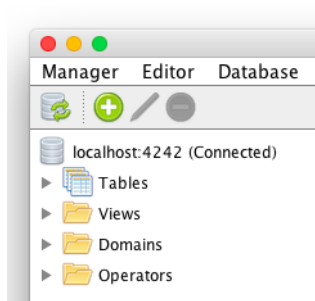
1. Cez hlavného menu aplikácie otvoríte dialógové okno otvorenia databázy. (*Manager* → *Open database*)



2. V okne sú prednastavené východzie parametre adresy (*Host name*) a portu (*Port*) na otvorenie lokálneho databázového servera. Úspešnosť spojenia je možné vopred otestovať tlačídlom *Test connection*.



3. Ak je spojenie úspešne vytvorené, načíta sa stromová štruktúra databázy.

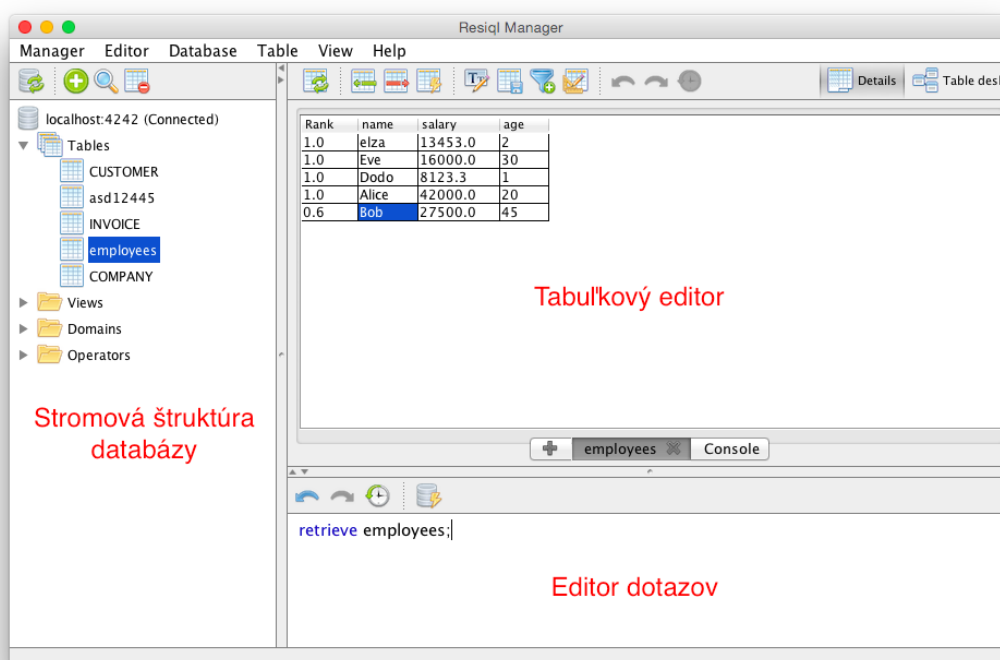


### 3.4 Popis hlavného okna

Aplikačné okno je rozdelené do troch panelov a hlavného menu. Menu umožňuje prístup k všetkým dôležitým funkciám aplikácie. Každý z panelov slúži na poskytovanie určitej časti funkcionality. Jednotlivé hlavné panely majú svoje vlastné panely nástrojov s podporovanými funkciami. Panely je možné zväčšovať alebo zmenšovať v závislosti na ostatných paneloch.

#### Hlavné panely:

- Stromová štruktúra databázy
- Tabuľkový editor
- Editor dotazov



Obr. 2: Rozdelenie panelov hlavného okna

Details o jednotlivých paneloch budú vysvetlené v nasledujúcich častiach príručky.

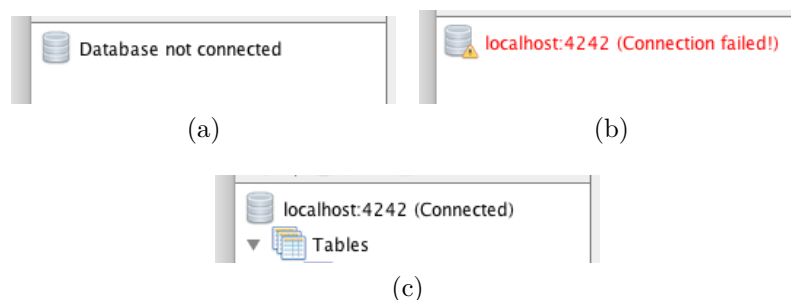
### 3.5 Stromová štruktúra databázy

Panel stromovej štruktúry databázy poskytuje základný prehľad o jej obsahu. Strom databázy je rozdelený do štyroch skupín:

- Tabuľky (*Tables*)
- Pohľady (*Views*)
- Domény (*Domains*)
- Operátory (*Operators*)

Koreň stromu predstavuje samotnú databázu s informáciou o jej aktuálnom stave (Obr. 3):

- Nepripojená databáza
- Nedostupná databáza
- Úspešne pripojená databáza




Obr. 3: Grafické zobrazenie aktuálneho stavu databázy

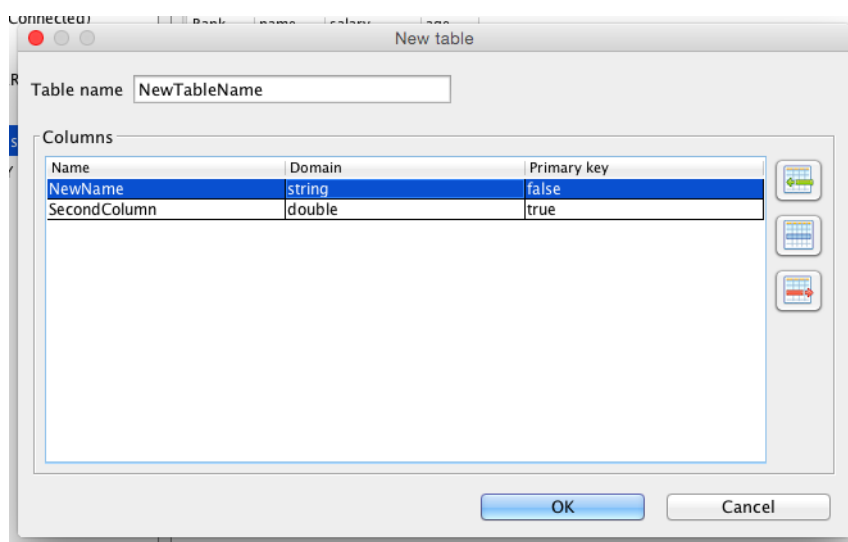
Panel nástrojov ponúka nasledujúce tlačidlá s možnosťami:

- Aktualizovať informácie o databáze 🔄.
- Tlačidlo + zobrazí kontextové menu s možnosťami na pridanie nového objektu (tabuľka, pohľad, doména alebo operátor).
- Ak je daný objekt v štruktúre označený, je možné upraviť ✎ prípadne zobrazit detaily objektu 🔍.
- Zmazanie objektu databázy 🗑️.

### 3.5.1 Tabuľky




Skupina tabuliek v stromovej štruktúre obsahuje zoznam užívateľských tabuliek databázy. Tabuľky je možné pridávať alebo mazať, ale momentálne nie je možné existujúce editovať z dôvodu limit aktuálnej verzie databázového systému. Namiesto editácie je dostupný náhľad na detail informácii tabuľky 🔍.



**Vytvorenie novej tabuľky** – Dialógové okno tvorby novej tabuľky (Obr. 4) otvoríte napríklad cez hlavné menu (*Database* → *New* → *Table*) alebo tlačidlom  v panely nástrojov.



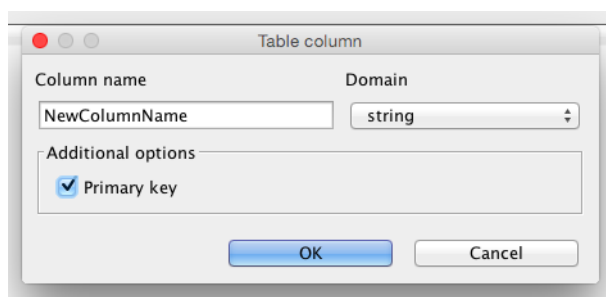
Obr. 4: Editor novej tabuľky

Možnosti editora novej tabuľky:

- Názov tabuľky (*Table name*) – Povinný, názov nesmie obsahovať medzery.
- Zoznam stĺpcov (*Columns*) – Obsahuje názov (*Name*), dátový typ (*Domain*) a informáciu či stĺpec patrí do primárneho kľúča tabuľky (*Primary key*). Povinný je minimálne jeden stĺpec.
- Tlačidlá naľavo od zoznamu stĺpcov slúžia na ich správu:
  - Pridanie nového stĺpca 
  - Úprava označeného stĺpca 
  - Zmazanie označeného stĺpca 

**Editor stĺpca tabuľky**  /  – Nový alebo existujúci stĺpec upravujete v okne editora stĺpca tabuľky (Obr. 5).



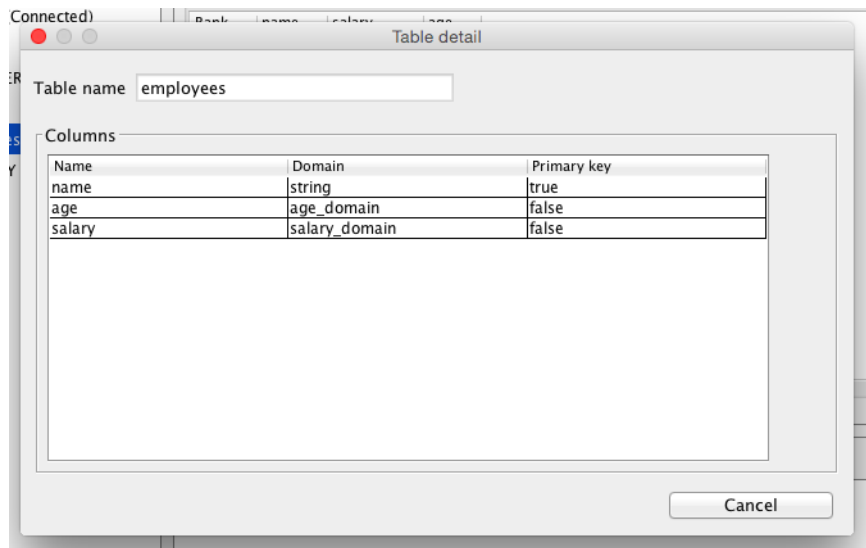


Obr. 5: Editor stĺpca tabuľky

Možnosti editora stĺpca tabuľky:

- Názov stĺpca (*Column name*) – Povinný bez medzier v názve.
- Dátový typ/doména (*Domain*) – Typ špecifikujúci aké dáta môžu byť v danom stĺpci ukladané.
- Dodatočne možnosti (*Additional options*) – Zahrnutie stĺpca do primárneho kľúča tabuľky.

**Náhľad tabuľky** 🔍 – je dostupný namiesto editácie. Zobrazuje zoznam stĺpcov tabuľky a ich parametre (Obr. 6).




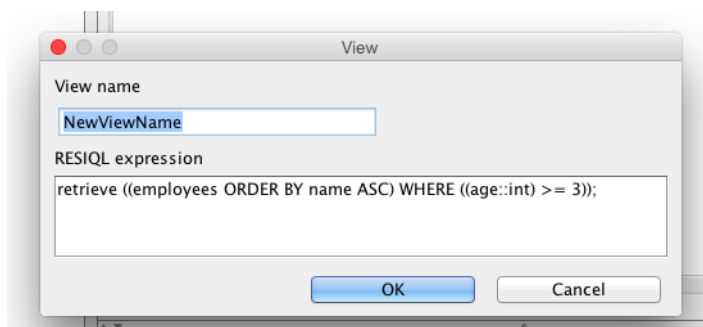
Obr. 6: Nahľad tabuľky

**Zmazanie tabuľky** 🗑️ - Zmazaniu predchádza kontrolná otázka, či naozaj chcete vybranú tabuľku odstrániť.

### 3.5.2 Pohľady

Skupina pohľadov (*Views*) v strome databázy. Funkcie na správu pohľadov sú dostupné cez panel nástrojov alebo hlavné menu. Pohľady je možné plne spravovať: vytvárať nové, upravovať alebo mazať existujúce.

**Vytvorenie nového pohľadu** – Dialógové okno nového pohľadu (Obr. 7) otvoríte cez hlavné menu (*Database* → *New* → *View*) alebo tlačidlom  v panele nástrojov.





Obr. 7: Editor pohľadu

Možnosti editora pohľadu:

- Názov pohľadu (*View name*) – Povinný údaj
- RESIQL výraz (*RESIQL expression*) – Výraz, ktorý vygeneruje tabuľku pohľadu.


V prípade, že neovládáte syntax jazyka RESIQL, nový pohľad si môžete navrhnuť pomocou tabuľkového editora. Postup ako navrhnuť tabuľku a uložiť ju ako pohľad nájdete v kapitole (3.7).

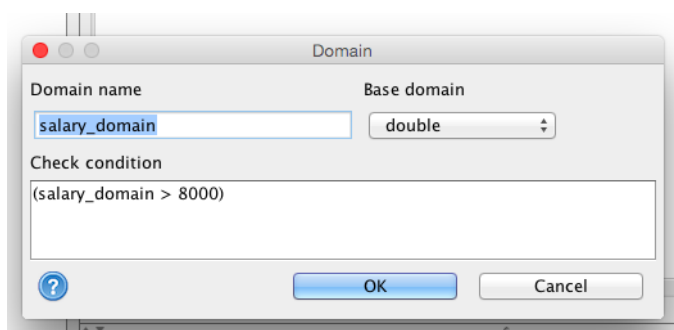
**Úprava pohľadu**  – Dialógové okno na úpravu existujúceho pohľadu je totožné s oknom na tvorbu nových pohľadov (Obr. 7).

**Zmazanie pohľadu**  – Zmazaniu predchádza kontrolná otázka, či naozaj chcete vybraný pohľad odstrániť.

### 3.5.3 Domény


Domény (*Domains*) je možné plnohodnotne spravovať.

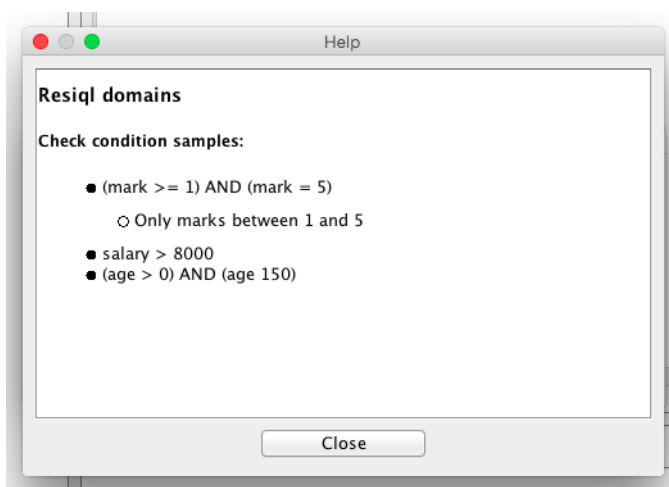
**Vytvorenie novej domény** – Dialógové okno novej domény (Obr. 8) otvoríte cez hlavné menu (*Database* → *New* → *Domain*) alebo tlačidlom  v panele nástrojov.




Obr. 8: Editor domény


Možnosti editora domény:

- Názov domény (*Domain name*) – Povinný údaj, musí byť jedinečný.
- Základná doména (*Base domain*) – Každá nová doména musí mať uvedený základný dátový typ (napríklad: *string*, *int*, *double*, *sys\_rank*).
- Kontrolná podmienka (*Check condition*) – Slúži na výber hodnôt alebo intervalu hodnôt, ktoré patria do novej domény.
- Nápoveda  – Zobrazí okno so vzorovými príkladmi kontrolných podmienok (Obr. 9).




Obr. 9: Nápoveda domény

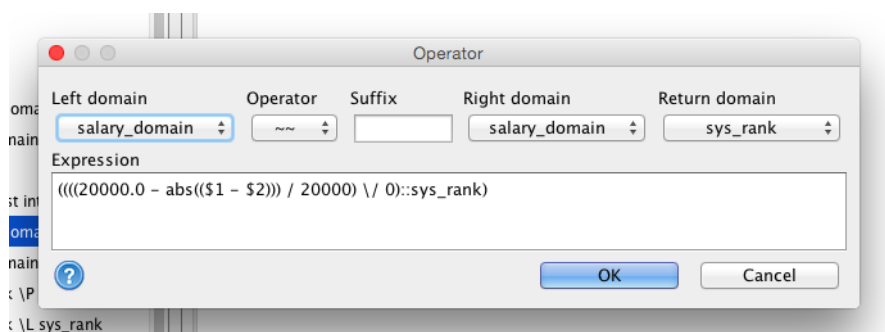
**Úprava domény**  – Dialógove okno na úpravu existujúcej domény je totožné s oknom na tvorbu novej domény (Obr. 8). Premenovanie je možné jedine vtedy, ak daná doména nie je súčasťou existujúcej tabuľky či operátora.

**Zmazanie domény**  – Zmazaniu predchádza kontrolná otázka, či naozaj chcete vybranú doménu odstrániť. Doménu nie je možné zmazať ak je súčasťou existujúcej tabuľky či operátora.

### 3.5.4 Operátory


Aplikácia ponúka správu binárnych operátorov (*Operators*), ktoré je možné plnohodnotne spravovať. V skupina operátorov obsahuje zoznam upraviteľných operátorov databázy.

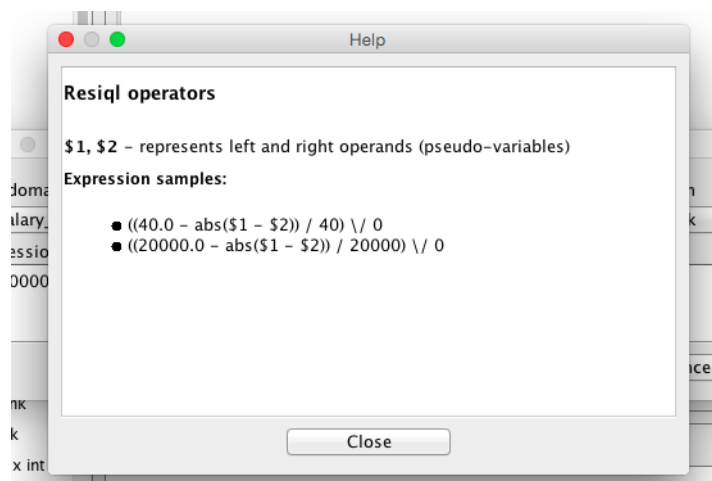
**Vytvorenie nového operátora** – Dialógové okno nového operátora (Obr. 10) otvoríte cez hlavné menu (*Database* → *New* → *Operator*) alebo tlačidlom  v panele nástrojov.




Obr. 10: Editor operátora


Možnosti editora operátora:

- Lavá, pravá a návratová doména (*Left, right and return domain*) – Volba zo všetkých dostupných domény, základné aj novo vytvorené.
- Operátor (*Operator*) – Znak operátora, ktorý je možné zvoliť zo sady podporovanej databázou.
- Prípona (*Suffix*) – Je voliteľná a slúži na odlišenie rôznych verzií toho istého operátora (napríklad použité odlišne domény operátora).
- Výraz (*Expression*) – Predstavuje RESIQL funkciu operátora, pomocou ktorej sa výraz vyhodnotí. Na vytvorenie funkcie je potrebné mať znalosti jazyka RESIQL.
- Nápoveda  – Zobrazí okno so vzorovými príkladmi funkcií operátorov (Obr. 11).



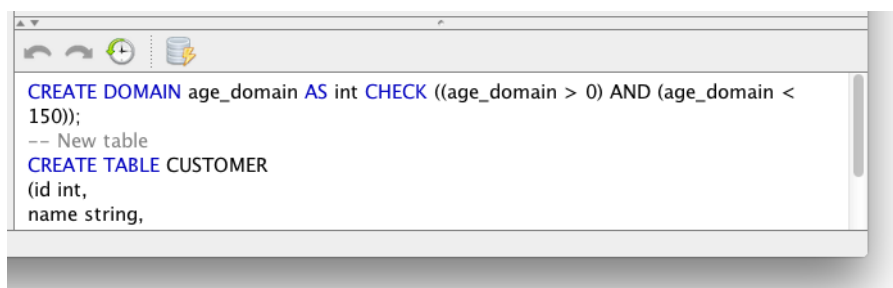
Obr. 11: Nápoveda operátora

**Úprava operátora**  – Dialógove okno na úpravu existujúcej domény je tožné s oknom na tvorbu nového operátora (Obr. 10).

**Zmazanie operátora**  – Zmazaniu predchádza kontrolná otázku, či naozaj chcete vybraný operátor odstrániť.

### 3.6 Editor dotazov

Tento textový editor slúži na zadávanie dotazov pre databázu. Podporuje nápovedu a farebné zobrazovanie syntaxe jazyka RESIQL. Dotazy z editora sú vyhodnocované databázou a o zobrazenie výsledkov sa stará tabuľkový editor popísaný v kapitole (3.7). Editor je primárne určený pre pokročilejších užívateľov so znalosťou jazyka.



Obr. 12: Editor dotazov

#### Vlastnosti editora:


- Farebné zvýrazňovanie slov
  - Modrá** slová jazyka RESIQL
  - Zelená** Textové reťazce
  - Sivá** Komentáre
  - Podčiarknuté červenou** Syntaktické chyby
  - Čierna** Ostatné slová
- Automatická zmena RESIQL operátora z ASCII slova na UNICODE znak. Výpis operátorov nájdete v tabuľke 1.

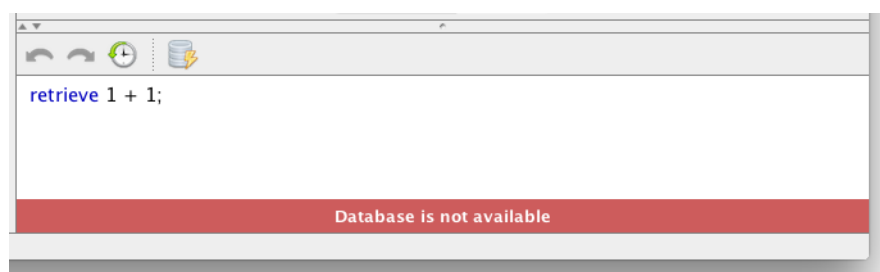
ASCII slovo	UNICODE znak
~~	≈
(+)	⊕
(*)	⊗
->	→
^	∧
v	∨
<>	≠
>=	≥
<=	≤
union	∪
intersect	∩

Tabuľka 1: ASCII – UNICODE operátory


- Zobrazovanie prípon RESIQL operátorov ako dolný index.
- Nápoveda kľúčových slov syntaxe jazyka RESIQL, názvov tabuliek a ich stĺpcov. Nápoveda po zadaní minimálne dvoch znakov navrhne najvhodnejšie dokončenie slova, zobrazí ho ako označený text. Na potvrdenie nápovedy slúžia klávesy *Enter*, *Tab* alebo *Šípky*, na prípadné zamietnutie nápovedy klávesy *Space*, *Backspace*.

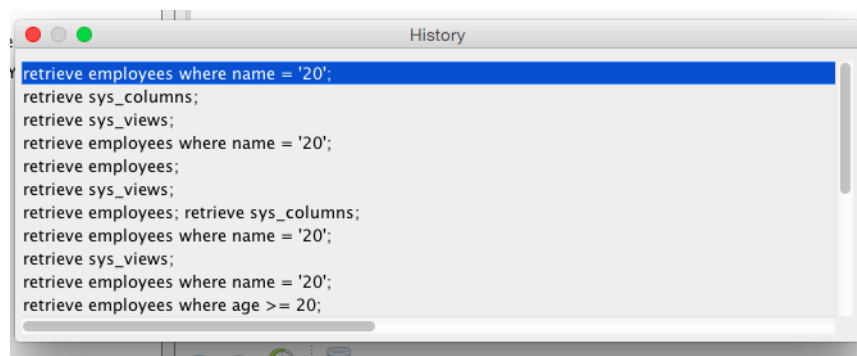
### Funkcie panela nástrojov:

- Vykonanie dotazu (*Execute query*)  – Po stlačení tlačidla alebo klávesy *Enter* je text v editore zmazaný a odoslaný do databázy na vyhodnotenie. Výsledok dotazu sa zobrazí v tabulkovom editore (3.7). Ak dotaz nie je možné vyhodnotiť, napríklad z dôvodu nedostupnej databázy, zobrazí sa chybová hláška a dotaz nebude zmazaný (Obr. 13).





Obr. 13: Editor dotazov s chybou

- História dotazov (*History*)  – Po stlačení tlačidla alebo kombinácie kláves *Ctrl + Space* sa zobrazí história posledných štyridsiatic databázou vyhodnotených dotazov. História je filtrovaná. Zobrazené sú len tie dotazy, ktorých začiatok je zhodný s aktuálnym textom v editore.



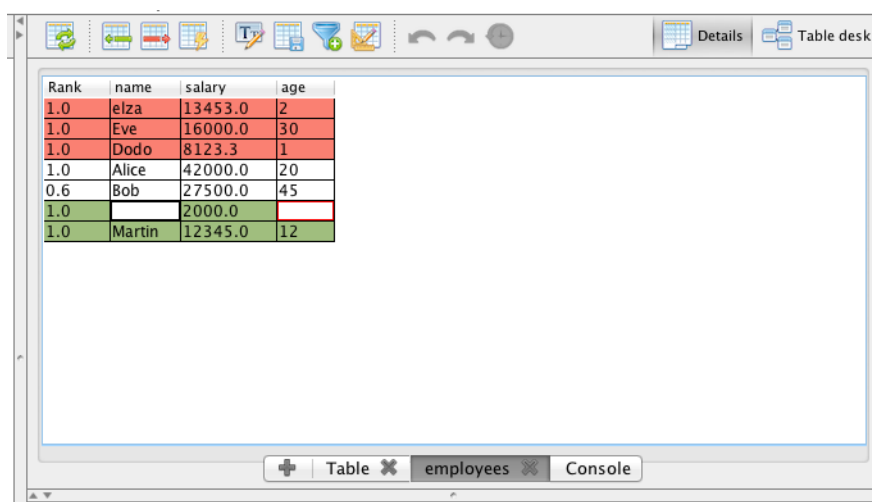
Obr. 14: História dotazov

Na doplnenie textu v editore dotazom z histórie sú určené klávesy *Enter* alebo *Space*, prípadne dvojklik myšou.

Rýchly prístup k položkám histórie umožňujú tlačidlá  predchádzajúci (*previous*) a  nasledujúci (*next*) dotaz.

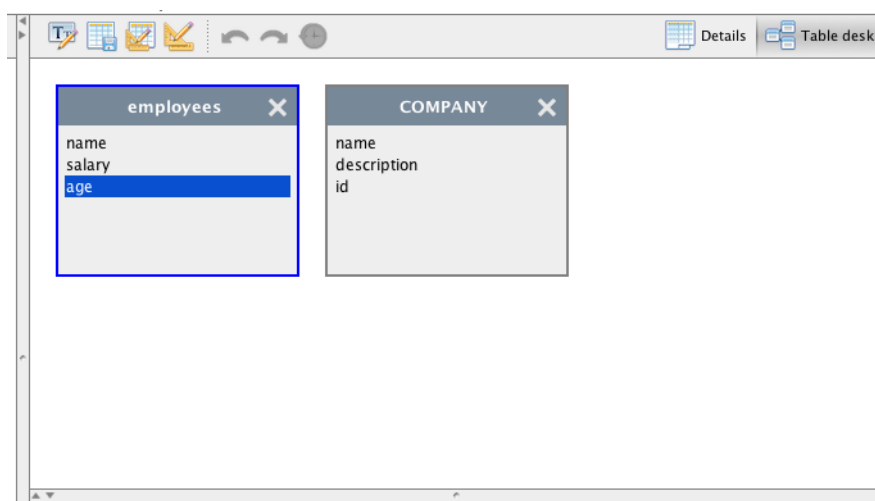
## 3.7 Tabuľkový editor

Tabuľkový editor je primárne určený na prácu s dátami databázy (tabuľky, pohľady). Ponúka rôzne funkcie na spracovanie dát v tabuľkách (napr. ich filtrácia, úpravy, atď.) a ďalších relačných operácií. Má na starosť konzolové výpisy výsledkov vyhodnotených RESIQL dotazov a textových informácií aplikácie. Umožňuje prácu v dvoch režimoch: režim detailov a režim tabuľkovej plochy.



Rank	name	salary	age
1.0	elza	13453.0	2
1.0	Eve	16000.0	30
1.0	Dodo	8123.3	1
1.0	Alice	42000.0	20
0.6	Bob	27500.0	45
1.0		2000.0	
1.0	Martin	12345.0	12

Obr. 15: Tabuľkový editor (Režim detailov)



employees
name
salary
age

COMPANY
name
description
id

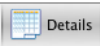


Obr. 16: Tabuľkový editor (Režim tabuľkovej plochy)

### 3.7.1 Režim detailov

Režim detailov (Obr. 15) je zameraný na prácu s dátami tabuliek, používanie filtrovacích a dizajnovacích operácií. Základom tohto režimu je ľubovoľný počet záložiek s tabuľkami, ktoré je možné pridávať a odoberať, a jedna špeciálna záložka určená na konzolové výpisy.



### Vlastnosti tabuľkového editora v režime detailov:

- Aktivovanie režimu tlačidlom  v pravej časti panela nástrojov alebo klávesovou skratkou  $Ctrl^5 + 1$ .
- Záložky  – Vždy je dostupná záložka konzoly a minimálne jedna záložka tabuľky. Tabuľkové záložky predstavujú tzv. kontajnery na zobrazenie tabuľkových dát. Na pridávanie nových tabuľkových záložiek slúži špeciálna záložka (tlačidlo)  alebo klávesová skratkou  $Ctrl + T$ . Záložky je možné zatvárať pomocou tlačidla umiestneného pri každej tabuľkovej záložke, prípadne použitím klávesovej skratky  $Ctrl + W$ . Aktívnej záložke sa prispôsobujú funkcie panela nástrojov.


### 3.7.2 Záložka konzoly

Konzola slúži ako textový výpis výsledkov vyhodnotených RESIQL dotazov a iných informácií aplikácie (Obr. 17).

#### Vlastnosti:

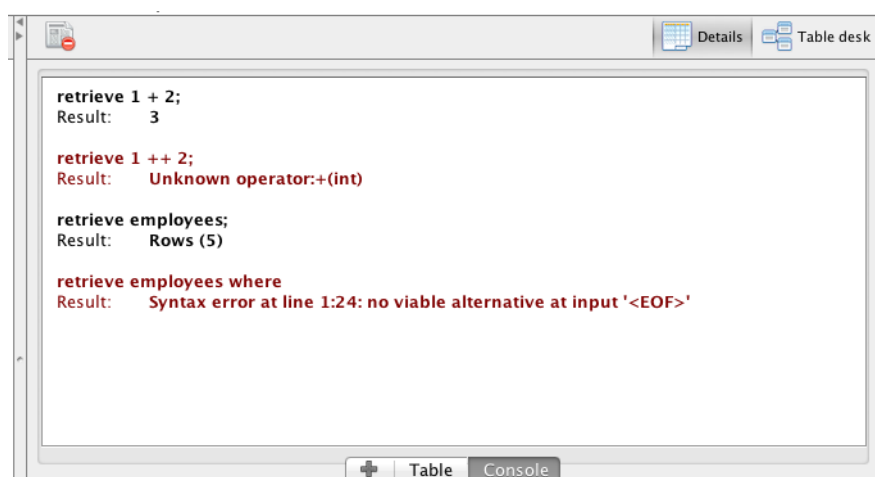
- Výpisy najčastejšie pochádzajú z dotazov vykonaných Editorom dotazov (3.6), prípadne z prídavných modulov.
- Záložka je aktivovaná automaticky ak RESIQL dotaz neobsahuje tabuľkové dáta.
- Jednotlivé výpisy sú formátované a farebne odlišné na základe typu výsledku vyhodnoteného dotazu. Ak došlo k chybe, výpis je zvýraznený červenou farbou (Obr. 17).

#### Funkcie panela nástrojov:

- Vyčistenie konzoly (*Clear console*)  – Odstráni všetky textové záznamy z konzoly.

---

<sup>5</sup>Tlačidlo *Cmd* v prípade používania operačného systému Mac OS X



Obr. 17: Záložka konzoly

### 3.7.3 Záložka tabuľky

Záložka predstavuje kontajner tabuľkových dát (Obr. 15).

#### Vlastnosti záložky:

- Tabuľku je možné načítať vykonaním RESIQL dotazu z Editoru dotazov alebo zo stromovej štruktúry databázy a to dvojklikom myšky na objekt tabuľky či pohľadu, prípadne klávesou *Enter*.
- Záložky tabuliek sú naplnené tabuľkovými dátami na základe nasledujúcich pravidiel:
  - Dáta aktívnej záložky sú prepísané novými.
  - Ak vykonaný dotaz vráti výsledok s viacerými tabuľkami, prvá tabuľka prepisuje tabuľku v aktívnej záložke. Pre ostatné tabuľky sa vytvorí nová tabuľková záložka.
  - Ak sa jedná o neupravenú základnú tabuľku databázy<sup>6</sup>, názov záložky je odvodený od jej názvu. V opačnom prípade sú jednotlivé záložky pomenované ako (*Table*) s poradovým číslom otvorenej tabuľky.

#### Funkcie panela nástrojov:

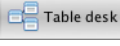
- Správa tabuľkových dát (3.7.5)
- Filtrovanie dát (3.7.6)
- Návrh tabuliek (3.7.7)
- Ostatné funkcie tabuľkového editora (3.7.9)

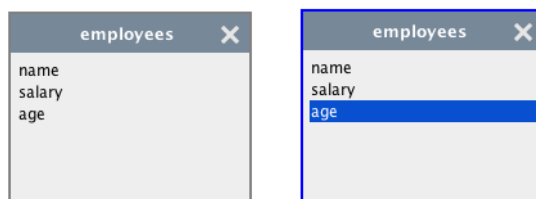
<sup>6</sup>Databázová tabuľka obsahujúca všetky stĺpce a záznamy.

### 3.7.4 Režim tabuľkovej plochy

Tabuľková plocha primárne slúži na dizajnovanie tabuliek a tvorbu nových použitím relačných operácií na existujúce tabuľky (Obr. 16).

**Vlastnosti tabuľkového editora v režime tabuľkovej plochy:**

- Aktivovaný pomocou tlačidla  v pravej časti panela nástrojov editora alebo použitím klávesovej skratky *Ctrl + 2* (Obr. 16).
- Tabuľky sú graficky zobrazené ako obdĺžniky obsahujúce názov danej tabuľky a zoznam jej stĺpcov.
- Každú tabuľku je možné zatvoriť po kliknutí na tlačidlo ✕.
- Dvojklik na tabuľku aktivuje režim detailov so zobrazenými dátami danej tabuľky
- Danú tabuľku označíme kliknutím priamo na ňu. Pre označenie viacerých tabuliek naraz je potrebné podržať stlačenú klávesu *Ctrl* a kliknúť myšou na dané tabuľky (Obr. 18).



(a) Tabuľka

(b) Označená tabuľka

Obr. 18: Reprezentácia tabuľky (tabuľková plocha)

**Funkcie panela nástrojov:**

- Tvorba relácií tabuliek (3.7.8)
- Ostatné funkcie tabuľkového editora (3.7.9)

### 3.7.5 Správa tabuľkových dát

Spravovať dáta tabuliek je možné len v prípade, že sa jedná o neupravenú databázovú tabuľku<sup>7</sup>. Ak je táto podmienka splnená, tabuľkový editor povolí funkcie na správu tabuľkových dát.

<sup>7</sup>Databázová tabuľka obsahujúca všetky stĺpce a záznamy.

Rank	name	salary	age
1.0	elza	13453.0	2
1.0	Eve	16000.0	30
1.0	Dodo	8123.3	1
1.0	Alice	42000.0	20
0.6	Bob	27500.0	45
1.0	NewName	4321.0	12
1.0			

Obr. 19: Správa tabuľkových dát

### Vlastnosti:

- Podpora pridania a odstránenia záznamov. Úprava databázou potvrdených záznamov nie je dovolená z dôvodu chýbajúcej podpory v [DBMS](#).
- Typová kontrola nových dát. Vkladaná hodnota sa musí zhodovať so základným dátovým typom stĺpca<sup>8</sup>.
- Pohyb medzi bunkami zabezpečujú klávesy: *Šípky*, *Tab*, *Enter*.
- Zápis a úprava obsahu bunky v novom riadku: *Ostatné klávesy*

### Funkcie panela nástrojov:

- Aktualizovať tabuľkové dáta (*Refresh table data*) – Načíta tabuľkové dáta z databázy. Všetky nepotvrdené zmeny budú stratené.
- Pridanie riadku (*Add table row*) – Tlačidlo alebo klávesová skratka *Ctrl + N* vytvorí nový prázdny riadok na konci tabuľky. Zelené pozadie bunky (Obr. 20a) signalizuje, že daná bunka obsahuje správnu hodnotu na základe bázeovej domény (viď kapitola Domény 3.5.3). Bunka označená červenou farbou okraja a písma (Obr. 20b) informuje užívateľa, že bunka nemôže byť prázdna alebo hodnota je nesprávneho dátového typu.
- Zmazanie riadku (*Delete table row*) – Tlačidlom alebo klávesovými skratkami *Cmd + Backspace* a *Delete*. Mazanie záznamov funguje na princípe označovania riadkov červenou farbou pozadia buniek (Obr. 20c). Nové riadky sú zmazané okamžite. Záznamy uložené v databáze sú reálne zmazané až po potvrdení zmien .
- Potvrdenie zmien (*Commit data*) – Tlačidlom alebo klávesovou skratkou *Ctrl + S* sú riadky označené ako zmazané a nové uložené do databázy. V prípade, že niektorá nová hodnota nespĺňa podmienku rámci svojho dátového typu sa zobrazí chybová hláška oznamujúca, že hodnota nepatrí do danej domény.

<sup>8</sup>Základné dátové typy: string, int, double, sys\_rank

0.6	Bob	27500.0	45
1.0	NewRow	23456.0	45

(a)

0.6	Bob	27500.0	45
1.0	NewRow	wrong value	45


(b)

1.0	Alice	42000.0	20
0.6	Bob	27500.0	45

(c)

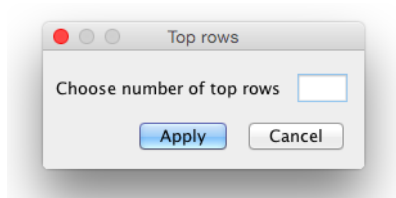
Obr. 20: Grafické označenie riadkov tabuľky

### 3.7.6 Filtrovanie dát

Filtrovacie funkcie sú zamerané výber konkrétneho záznamu či intervalu záznamov z tabuľkových dát. Funkcie sú dostupné pod tlačidlom  v režime detailov s aktívnou záložkou tabuľky s dátami.

#### Najlepších $X$ riadkov (*Top $X$ rows*)

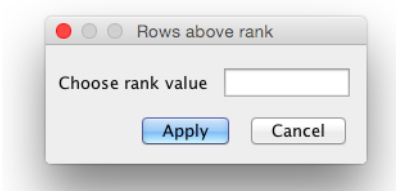
Zobrazí  $X$  najlepších záznamov tabuľky (Obr. 21).



Obr. 21:  $X$  najlepších riadkov

#### Riadky nad ohodnotením (*Rows above rank*)

Zobrazí záznamy tabuľky, ktoré majú ohodnotenie (*Rank*) rovné a vyššie ako je špecifikované v dialógovom okne (Obr. 22).





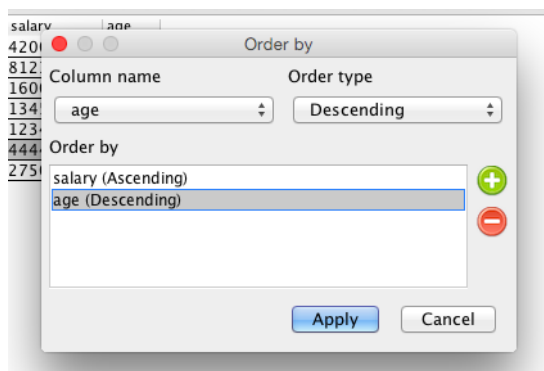
Obr. 22: Riadky nad ohodnotením

#### Zoradiť podľa (*Order by*)

Funkcia dovoľí nastaviť poradie záznamov na základe stĺpcov tabuľky (Obr. 23). Okno obsahuje zoznam stĺpcov a ich typu radenia.

Použitie:

- Zvolenie názvu stĺpca (*Column name*)
- Zvolenie typu radenia (*Order type*)
- Možnosť pridať  alebo odobrať  stĺpec z radenia



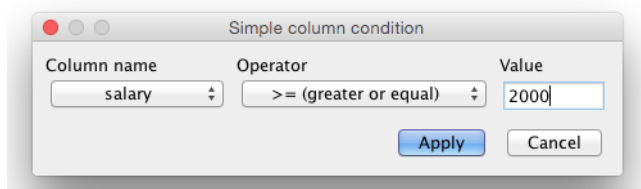
Obr. 23: Radenie podľa

### Jednoduchá podmienka (*Simple condition*)

Jednoduchá podmienka dovoľí použitie jedného operátora na vybraný stĺpec tabuľky s voliteľnou hodnotou (Obr. 24).

Použitie:

- Zvolenie názvu stĺpca tabuľky (*Column name*).
- Zvolenie operátora (*Operator*) – Obsahuje štandardné operátory porovnávania a vlastné nové operátory.
- Napísanie hodnoty (*Value*).




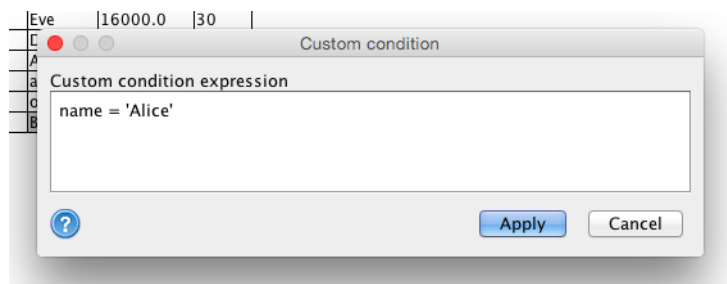
Obr. 24: Jednoduchá podmienka

### Vlastná podmienka (*Custom condition*)

Funkcia slúži na špecifikovanie vlastnej podmienky. Užívateľ musí mať skúsenosti s jazykom RESIQL (Obr. 25).


Použitie:

- Do textového poľa je zadávaná podmienka po kľúčovom slove *where* z jazyka RESIQL.
- Príklady podmienok sú dostupné v nápovede .



Obr. 25: Vlastná podmienka

### 3.7.7 Návrh tabuliek

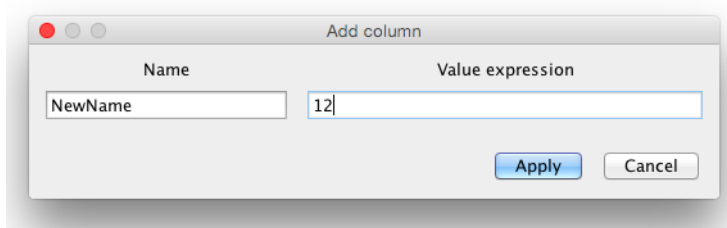
Pomocou týchto funkcií je možné pridávať, odoberať alebo premenovať stĺpce tabulky. Dostupné sú v oboch režimoch editora pod tlačidlom  v panely nástrojov.

#### Pridanie stĺpca (Add column)

Zobrazí dialógové okno na zadanie dodatočných informácií potrebných pre vytvorenie nového stĺpca (Obr. 26).

Použitie:

- Vyplnenie názvu stĺpca (Column name)
- Hodnota alebo výraz (Value expression) – Výraz môže byť hodnotou alebo RESIQL výrazom, prípadne agregáčnou funkciou



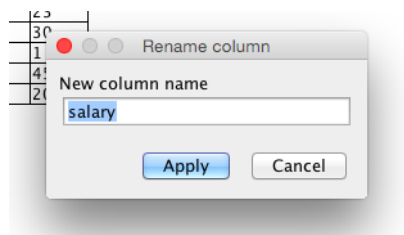
Obr. 26: Pridanie stĺpca

#### Premenovanie stĺpca (Rename column)

Zobrazí dialógové okno premenovania (Obr. 27) na základe označenej bunky v stĺpci tabulky.

Použitie:

- Vyplnenie nového názvu stĺpca (New column name)



Obr. 27: Premenovanie stĺpca

### Zmazanie stĺpca (Delete column)

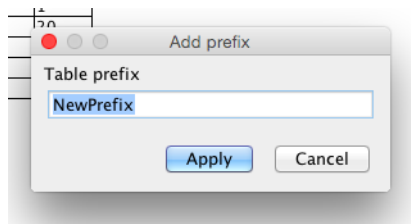
Na základe označenej bunky v stĺpci tabuľky je zmazaný daný stĺpec.

### Pridanie predpony (Add prefix)

Zobrazí dialógové okno na pridanie predpony pre označenú tabuľku (Obr. 28).


Použitie:

- Vyplnenie predpony tabuľky



Obr. 28: Pridanie predpony

### 3.7.8 Tvorba relácii tabuliek

Pomocou týchto funkcií je možné vytvárať nové tabuľky použitím relačných operácií na označené tabuľky. Dostupné sú len v režime tabuľkovej plochy editora pod tlačidlom  v panele nástrojov. Editor ponúka niekoľko základných funkcií a možnosť použiť vlastný operátor.

#### Zoznam základných funkcií:

- Kartézsky súčin (*Cross join*)
- Prirodzený súčin (*Natural join*)
- Silný prienik (*Strong intersection*)
- Prienik (*Intersection*)
- Silné zjednotenie (*Strong union*)
- Zjednotenie (*Union*)



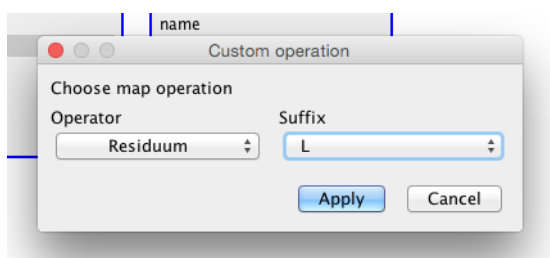
- Rozdiel (*Minus*)
- Zvyšok (*Residuum*)

### Vlastný operátor (Customize operator)

Zobrazí sa dialógové okno, ktoré umožňuje použitie rozšírených verzií základných funkcií (Obr. 29).

Použitie:




- Zvolenie operátora funkcie
- Zvolenie prípony (*Suffix*) operátora

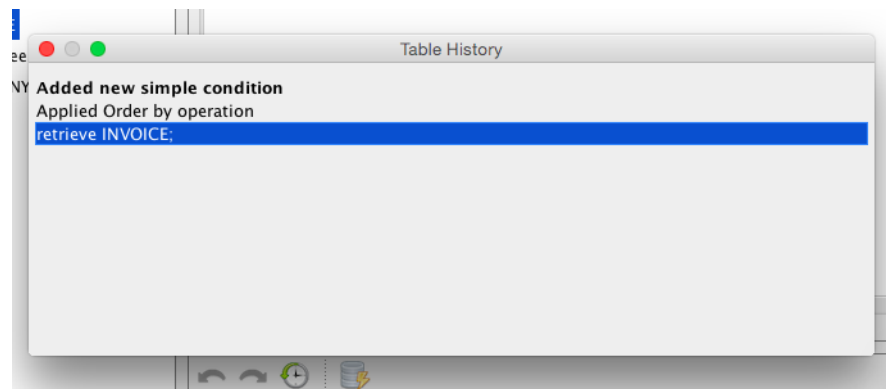


Obr. 29: Vlastný operátor

### 3.7.9 Ostatné funkcie tabuľkového editora



Nasledujúce funkcie panela nástrojov sú dostupné v oboch režimoch editora:

- Zobrazenie dotazu (*Show query*)  – Zobrazí dotaz označenej tabuľky v Editore dotazov (3.6).
- Uložiť ako pohľad (*Save as view*)  – Zobrazí dialógové okno na vytvorenie nového pohľadu označenej tabuľky (viď 3.5.2).
- História tabuľky (*Table history*)  – Stlačením tlačidla alebo klávesovej skratky *Shift + Ctrl + Space* sa otvorí okno so zoznamom histórie tabuľky (Obr. 30). Každá tabuľka otvorená v editore si ukladá históriu zmien, ktoré vznikli použitím návrhových, filtrovacích funkcií a prepísaním tabuľkových dát inými. Aktívny stav tabuľky je v zozname zmien označený tučným písmom.



Obr. 30: História zmien tabuľky

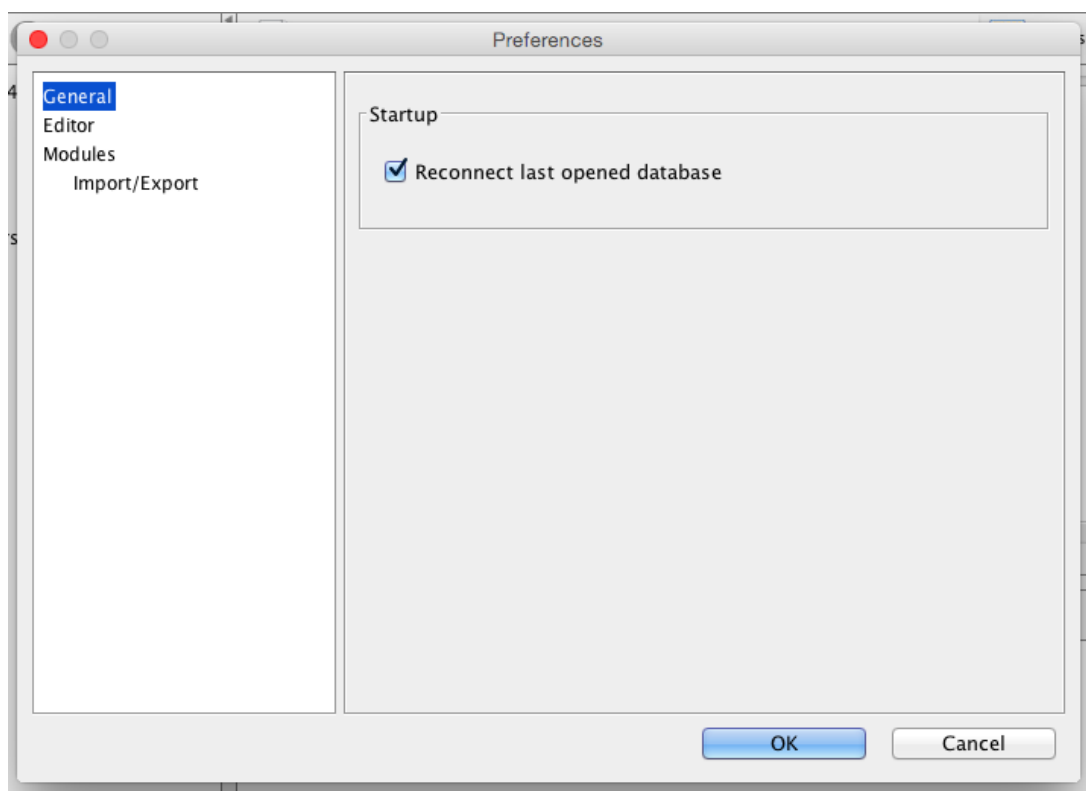
Na zobrazenie tabuľky z histórie sú určené klávesy *Enter* alebo *Space*, prípadne dvojklik myšou.

Rýchly prístup k položkám histórie umožňujú tlačidlá  vrátiť späť zmenu (*Undo change*) a  obnoviť zmenu (*Redo change*) tabuľky.

### 3.8 Nastavenia aplikácie

Nastavenia aplikácie sú dostupné cez hlavné menu (*File* → *Preferences*).

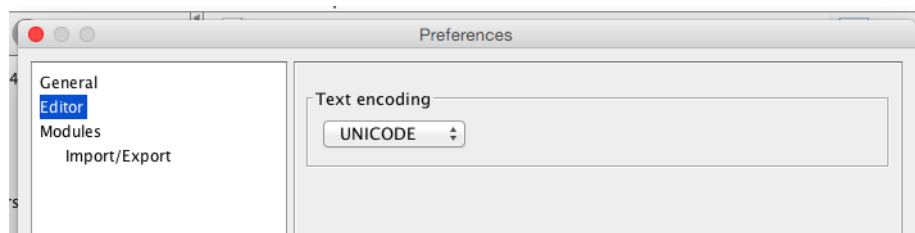
V aktuálnej verzii existuje dialógové okno s minimálnym množstvom možností (Obr. 31). Okno sa skladá z dvoch panelov. Ľavý panel zobrazuje zoznam kategórií nastavení, pravý panel konkrétne nastavenia spadajúce do danej kategórie.



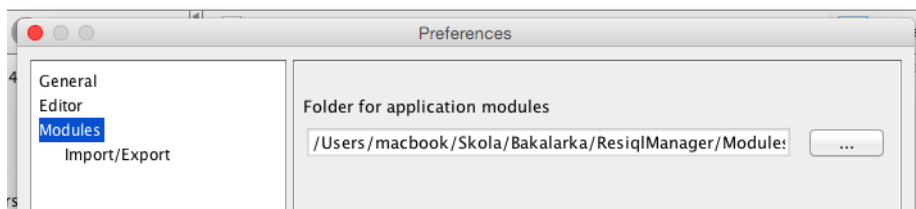
Obr. 31: Okno nastavení aplikácie

Dostupné nastavenia vrámci kategórií:

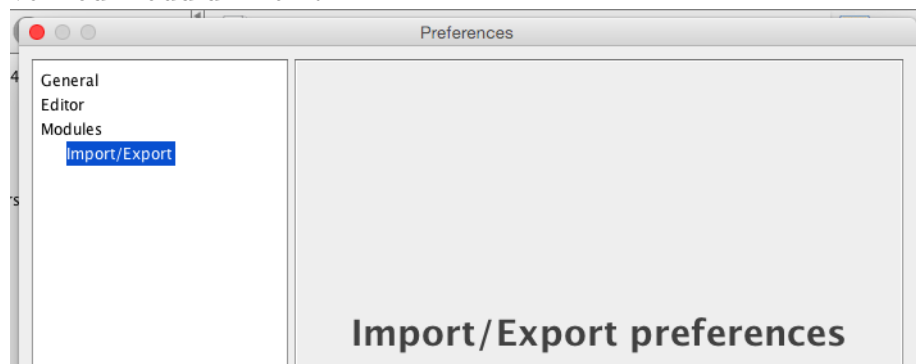
- **Všeobecné (General)** - Možnosť pri spustení aplikácie automaticky pripojiť poslednú známu použitú databázu (Obr. 31).
- **Editor** - Možnosť zmeniť používanú znakovú sadu UNICODE alebo ASCII v textovom editore (3.6).



- **Moduly (Modules)** - Obsahuje informácie o ceste k zložke, v ktorej sú uložené prídavné moduly. Možnosť tlačidlom zobrazíť danú zložku. Ako podkategórie v ľavom panele zobrazuje aktívne moduly.



- **Import/Export** - Predstavuje nastavenia načítaného prídavného modulu. Súčasnosti modul neponúka žiadne možnosti a preto namiesto nastavení je zobrazený len text. To sa ale môže v budúcnosti s novou verziou modulu zmeniť.

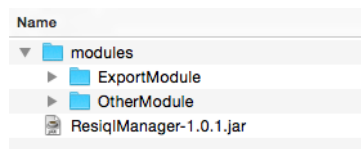


### 3.9 Prídavné moduly

Prídavné moduly slúžia na rozšírenie funkcionality aplikácie. Moduly sú aplikáciou načítané automaticky pri spustení z prednastavenej zložky s názvom *modules*. Táto zložka sa nachádza pri spúšťanej aplikácii (Obr. 32).

Jednotlivé moduly sú od seba odlišené zložkou, v ktorej sa nachádzajú súbory a dáta modulu.

Vybraný modul stačí „pridať do/odobrať zo“ zložky *modules* a reštartovať aplikáciu.



Obr. 32: Umiestnenie prídavných modulov

Postup tvorby vlastných modulov nájdete v Technickej príručke, kapitola (4.5).

## 4 Technická príručka

Príručka slúži na oboznámenie vývojára so základnou štruktúrou a princípmi aplikácie *RESIQL Manager*. Po prečítaní by mal byť schopný orientovať sa v zdrojových kódoch a pridávať novú funkcionality. Príručka obsahuje informácie o použitých nástrojoch a materiáloch, ktoré boli použité pri vývoji aplikácie. Popísaná je základná štruktúra projektov a organizácia jednotlivých zdrojových kódov. Následne vysvetlená hierarchia dôležitých riadiacich tried a rozhraní aplikácie ako aj ich použitie a princípy fungovania. Záver príručky je venovaný vytvoreniu jednoduchého prídavného modulu aplikácie.

### 4.1 Vývojárske nástroje a materiály

Základné technické informácie o aplikácii:

- Programovací jazyk – **Java**
- Platforma – **Java Standard Edition (Java SE)**
- Verzia behového prostredia – **Java Runtime Environment 8 (JRE 8)**
- Použité vývojové prostredie – **Eclipse IDE Luna**
- Licencia projektov – **GNU General Public License, version 2 (GPL-2.0)**<sup>9</sup>

Použité knižnice tretej strany:

- **resiq client driver a parser** – RESIQL knižnice slúžiace na komunikáciu s databázou a prácu s dotazmi. Vyvíjané Katedrou informatiky na Prírodovedeckej fakulte Univerzity Palackého v Olomouci a hlavným správcom je Mgr. Petr Krajča, Ph.D.  
Viac informácií na: <http://phoenix.inf.upol.cz/krajcap/data/resiq-0.3.3.3/>
- **Log4j** – Logovacia služba aplikácie.  
Viac informácií na: <http://logging.apache.org/log4j/1.2/>
- **x-stream** – Knižnica na ukladanie objektov tried do súborov v XML formáte. Viac informácií na: <http://xstream.codehaus.org>
- **common-io** – Rozšírenie funkcionality na prácu so súborovým systémom.  
Viac informácií na: <http://commons.apache.org/proper/commons-io/>

Použité zdrojové kódy tretej strany:

---

<sup>9</sup><http://opensource.org/licenses/GPL-2.0>

- **DeepCloner** – Slúži na úplne klonovanie objektov. Viac informácií na: <http://alvinalexander.com/java/java-deep-clone-example-source-code>
- **TableColumnAdjuster** – Trieda má na starosť automatické nastavovanie šírky stĺpcov `JTable`. Viac informácií na: <http://tips4java.wordpress.com/2008/11/10/table-column-adjuster/>
- **WrapLayout** – Slúži ako rozšírenie `FlowLayout` triedy. Viac informácií na: <http://tips4java.wordpress.com/2008/11/06/wrap-layout/>

Iné materiály:

- Ikony použité v aplikácii: <http://www.fatcow.com/free-icons>

## 4.2 Aplikácia RESIQL Manager

Pri návrhu štruktúry projektov bol zohľadnený návrhový vzor Model-View-Controller<sup>10</sup>. Aplikácia je rozdelená do dvoch projektov. Hlavný projekt s jadrom aplikácie `ResiqlManager` a knižnica `ModuleLibrary` s rozhraniami a triedami potrebnými pre komunikáciu hlavnej aplikácie s prídavnými modulmi.

Kapitola popisuje štruktúru projektu hierarchiu hlavných ovládačov aplikácie a ako pracovať s niekoľkými základnými komponentami a triedami aplikácie.

### 4.2.1 Štruktúra projektu ResiqlManager

Popis jednotlivých dôležitých zložiek projektu.

**components** – Komponenty používané v rôznych častiach aplikácie

**database** – Poskytovateľ prístupu k databáze

**helper** – Komponenta nápovedy

**moduleprovider** – Poskytovateľ rozhraní pre prídavné moduly

**resiqleditor** – Základ textového RESIQL editora dotazov

**preferences** – Poskytovateľ nastavení aplikácie

**main** – Hlavný inicializačný zdrojový kód, konštanty a globálne premenné aplikácie

**resources** – Materiály používané aplikáciou (Html texty, ikony, ...)

**utilities** – Pomocné triedy a triedy tretích strán

**views** – Všetky okná aplikácie

**about** – Okno o základných informáciách aplikácie

<sup>10</sup><http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

**creator** – Okná na tvorbu nových objektov databázy  
**domain** – Správa domén  
**operator** – Správa operátorov  
**table** – Správa tabuliek  
**view** – Správa pohľadov  
**main** – Hlavné okno aplikácie  
**menubar** – Hlavné menu  
**queryeditor** – Editor dotazov (3.6)  
**tableeditor** – Tabuľkový editor (3.7)  
**treeview** – Stromový pohľad na databázovú štruktúru (3.5)  
**opendatabase** – Okno pripojenia databázy  
**preferences** – Okno Nastavení aplikácie

#### 4.2.2 Inicializačný proces aplikácie

Aplikácia je spustená z triedy `MainApp`. Táto trieda má na starosť inicializačný proces celej aplikácie. Jednotlivé časti sú spúšťané v nasledujúcom poradí:

1. Logovacia služba
2. Nastavenia aplikácie
3. Poskytovateľ databázy
4. Hierarchia základných ovládačov (*Controllers*) všetkých častí aplikácie
5. Ovládač prídavných modulov (`ModuleController`)

**Logovacia služba** využíva možnosti knižnice `Log4j`<sup>11</sup>. Logovanie je nastavené na zapisovanie záznamov do súboru, ktorý je uložený v zložke *ResiqlManager* umiestnenej medzi užívateľskými aplikačnými dátami<sup>12</sup>.

**Nastavenia aplikácie** sú pri spustení načítané zo súboru. K tomu slúži knižnica `x-stream`, ktorá transformuje java objekty do XML súboru. Vedľa užívateľských nastavení sú ukladané aj nastavenia prídavných modulov prostredníctvom poskytovateľa `PreferencesProvider`.

**Poskytovateľ databázy** je vytvorený pri spustení aplikácie hlavný objekt pre komunikáciu s databázou `ResiqlDatabase`.

<sup>11</sup><http://logging.apache.org/log4j/1.2/>

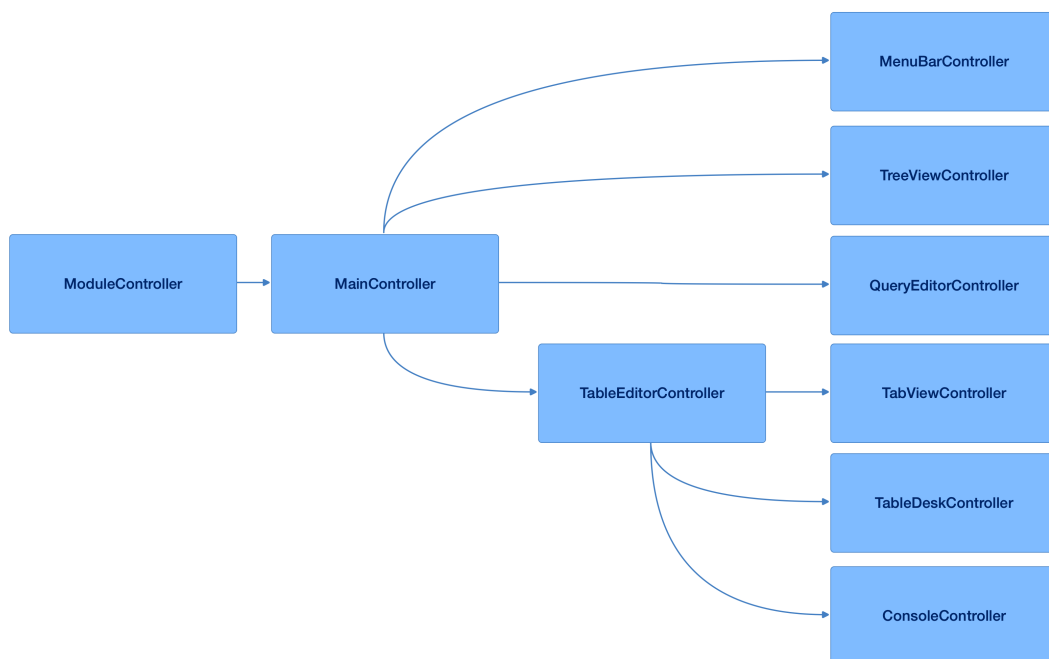
<sup>12</sup>Umiestnenie užívateľských aplikačných dát závisí na operačnom systéme.

**Hierarchia základných ovládačov** predstavujú kostru aplikácie, kde každý ovládač (*Controller*) má na starosť istú časť funkcionality. Každý z týchto základných ovládačov implementuje rozhranie poskytovateľa (*Provider*) funkcionality pre prídavné moduly. Bližšie informácie o jednotlivých ovládačov sú popísane v kapitole (4.2.3).

**Ovládač prídavných modulov** je vytvorený ako posledný, aby správne nastavil prístup k jednotlivým ovládačom aplikácie pre prídavné moduly. Následne prebehne proces detekcie a inštalácie jednotlivých prídavných modulov. Ovládač vyhľadáva v podzložkách zložky (*./modules*) java archívy<sup>13</sup>, ktorých hlavná trieda implementuje inštalčné rozhranie `ModuleBase`. Nad každým nájdeným prídavným modulom spustí inštaláciu.

### 4.2.3 Hierarchia základných ovládačov

Každý ovládač má na starosť istú časť funkcionality aplikácie. Podrobné informácie o jednotlivých metódach sú dostupné v priloženej java dokumentácii na priloženom CD (*bin/ResiqIManager/Java.Doc*). Zoznam základných ovládačov:



Obr. 33: Hierarchia základných ovládačov

**MainController** – Základný ovládač rozdeľuje hlavné okno aplikácie medzi ostatné ovládače (`MenuBarController`, `QueryEditorController`, `TreeViewController`, `TableEditorController`).

<sup>13</sup>Java archív predstavuje súbor s príponou \*.jar



**ModuleController** – Ovládač sprístupňuje funkcionality celej aplikácie pre prídavné moduly.

**TableEditorController** – Tabuľkový editor má na starosť tri ďalšie ovládače (*TableViewController*, *TableDeskController*, *ConsoleController*). Okrem týchto ovládačov spravuje zoznam objektov *TableObject*, ktoré obsahujú tabuľkové dáta zobrazených tabuliek.

**TableViewController** – Záložkový ovládač tabuliek sa stará o prácu s tabuľkovými dátami.

**ConsoleController** – Záložkový ovládač určený na prácu s konzolou.

**TableDeskController** – Ovládač sa stará o funkcionality tabuľkovej plochy.

**QueryEditorController** – Ovládač editora dotazov.

**TreeViewController** – Ovládač zobrazenia stromovej štruktúry databázy.

**MenuBarController** – Ovládač hlavného menu aplikácie.

## 4.3 Knižnica *ModuleLibrary*

Na komunikáciu modulu s aplikáciou je nutné použiť knižnicu *ModuleLibrary*, ktorú nájdete v CD prílohe v adresári (*bin/Module/ModuleLibrary.jar*). Jednotlivé triedy a rozhrania sú popísané v java dokumentácii na priloženom CD (*bin/ModuleLibrary/JavaDoc*).

### 4.3.1 Štruktúra projektu *ModuleLibrary*

Popis jednotlivých dôležitých zložiek projektu.

**upol.cz.module** – Základne prístupové rozhrania a spoločné triedy viacerých rozhraní

**database** – Databázová komponenta

**menubar** – Hlavné menu aplikácie

**preferences** – Prístup k nastaveniam aplikácie

**progressworker** – Komponenta pre dlhotrvajúce operácie.

**textovyyeditor** – Editor dotazov (3.6)

**tabulkovyyeditor** – Tabuľkový editor (3.7)

**treeview** – Stromová štruktúra databázy (3.5)

## 4.4 Použitie vybraných komponent a tried

Kapitola popisuje príklady použitia vybraných dôležitých častí aplikácie.

#### 4.4.1 Práca s databázou

Prístup k databáze zabezpečuje komponenta s hlavnou triedou `ResiqlDatabase` prípadne jej rozhranie `DatabaseProvider` pre prídavné moduly.

Získať poskytovateľa je možné dvoma spôsobmi. V hlavnej aplikácii pomocou volania z objektu `ResiqlDatabase.getInstance()`; alebo prostredníctvom metódy `ModuleProvider.getDatabaseProvider()`;

#### Príklad vykonania dotazu nad databázou:

Zdrojový kód nastaví spojenie s databázou a vykoná nad ňou jednoduchý RESIQL dotaz (Kód 1).

```
1  @Override
2  public void installModule(ModuleProvider moduleProvider) {
3      // Získanie prístupu k databazovému poskytovateľu
4      DatabaseProvider databaseProvider = moduleProvider.
5          getDatabaseProvider();
6      // Nastavenie spojenia s databázou
7      databaseProvider.setConnection("localhost", 4242);
8
9      try {
10         // RESIQL dotaz
11         String query = "retrieve 1 + 1;";
12         query = "retrieve sys_columns;";
13         query = "retrieveeeee";
14         query = "retrieve sys_columns; retrieve 1 + 1;";
15         // Vykonanie dotazu nad databázou
16         QueryResult queryResult = databaseProvider.executeQuery(query);
17         if (queryResult instanceof ValueQueryResult) {
18             // Hodnotový výsledok
19         } else if (queryResult instanceof TableQueryResult) {
20             // Tabulkové data
21         } else if (queryResult instanceof ErrorQueryResult) {
22             // Chybový výsledok
23         } else if (queryResult instanceof MultiQueryResult) {
24             // Výsledky viacerých dotazov
25         }
26     } catch (DatabaseException e) {
27         e.printStackTrace();
28     }
```

Zdrojový kód 1: Vykonanie dotazu nad databázou

#### 4.4.2 Použitie ProgressWorker komponenty

Komponenta slúži na spustenie algoritmu v novom samostatnom vlákne. Pričom počas behu je zobrazené dialógové okno s informáciou o tom čo sa práve vykonáva. ProgressWorker poskytuje možnosť nastaviť oneskorenie zobrazenia dialógového okna. Pre krátko trvajúce algoritmy sa okno nezobrazí. Komponenta je vlákno bezpečná (*Thread-Safe*).

Objekt ProgressWorker je možné získať dvoma spôsobmi. Priamym vytvorením objektu triedy alebo v prípade prídavného modulu cez ModuleProvider a metódou getProgressWorkerProvider.

##### Príklad použitia ProgressWorker komponenty:

Kód nastaví objekt ProgressWorker s 300 milisekundovým oneskorením zobrazenia dialógového okna s úlohou na pozadí a výpisom po skončení úlohy (Kód 2).

```
1 public static void main(String[] args) {
2     ProgressWorker worker = new ProgressWorker("Title", "Message",
3         300, null);
4     worker.setBackgroundTaskListener(new BackgroundTaskListener() {
5         @Override
6         public Object doWork(ProgressWorker worker) throws Exception {
7             Thread.sleep(2000);
8             // Algoritmus na pozadii
9             return "Work is done";
10        }
11
12        @Override
13        public void workDone(ProgressWorker worker) {
14            try {
15                // Ziskanie vysledku
16                Object result = worker.get();
17                System.out.println(result);
18            } catch (InterruptedException | ExecutionException e) {
19                e.printStackTrace();
20            }
21        }
22    });
23    worker.execute();
24 }
```

Zdrojový kód 2: Použitie ProgressWorker komponenty

## 4.5 Vytvorenie prídavného modulu

Prídavné moduly majú slúžiť ako rozšírenie funkcionality aplikácie *RESIQL Manager*. Komunikácia medzi hlavnou aplikáciou a modulmi prebieha cez jej rozhrania ovládačov (*Controllers*) jednotlivých častí. Počas inicializácie trieda zodpovedná za prídavné moduly ich identifikuje a spustí inštaláciu jednotlivých modulov s prístupovými rozhraniami do hlavnej aplikácie. Každý modul sa teda sám stará o to akým spôsobom sa do nej integruje a aké zdroje využije. Dostupné funkcie pre prídavné moduly sú popísané v kapitolách (4.2, 4.3).

Nasledujúci príklad popisuje implementáciu jednoduchého modulu HelloWorld, od založenia projektu až po vytvorenie použiteľného java súboru. Návod nepopisuje detaily týkajúce používania programu Eclipse. Programátor by mal poznať program.

Na začiatok je potrebné mať pripravený program Eclipse IDE pre java vývoj a knižnicu *ModuleLibrary.jar* zo zložky (*bin/ModuleLibrary*) priloženého CD.

### Postup:

1. Založiť nový java projekt v programe Eclipse IDE
2. Pridať do projektu knižnicu *ModuleLibrary.jar*
3. Vytvoriť novú triedu *Main.java* s vygenerovanou main metódou v balíku helloworldmodule.
4. Rozšíriť *Main.java* o rozhranie `ModuleBase` z `ModuleLibrary` (kód 3)

```
1 package helloworldmodule;
2
3 import upol.cz.module.ModuleBase;
4 import upol.cz.module.ModuleProvider;
5
6 public class Main implements ModuleBase {
7
8     public static void main(String[] args) {
9     }
10
11     @Override
12     public void installModule(ModuleProvider moduleProvider) {
13     }
14
15     @Override
16     public void unistallModule() {
17     }
18 }
```

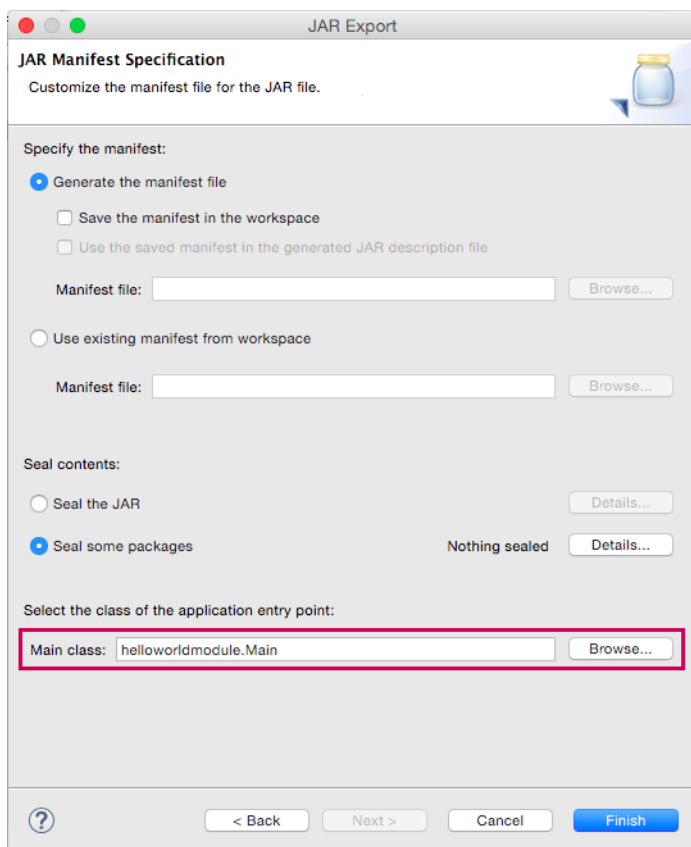
Zdrojový kód 3: Vytvorená trieda Main.java

5. Metóda `installModule` je volaná hlavnou aplikáciou ako prvá. Následne je možné používať prostredníctvom `ModuleProvider` rozhrania všetky dostupné možnosti aplikácie.
6. Pridanie kódu do `installModule` metódy na výpis textu Hello world from my module v konzole hlavnej aplikácie (Kód 4)

```
1  @Override
2  public void installModule(ModuleProvider moduleProvider) {
3      // Access to Table editor
4      TableEditorProvider tableEditorProvider = moduleProvider.
        getTableEditorProvider();
5      // Access to Console tab
6      ConsoleProvider consoleProvider = tableEditorProvider.
        getConsoleProvider();
7      // Write text to console
8      consoleProvider.writeToConsole("\nHello world from my module!\n\n
        ");
9  }
```

Zdrojový kód 4: Zapísanie textu do konzoly

7. Zostaviť java archív modulu. Dôležité je nastaviť Main triedu ako hlavnú v java archíve.



8. Použitie zostaveného modulu je popísané v kapitole (3.9)

## Záver

Na základe vlastností RESIQL databázy a požiadavok vedúceho práce bola vyvinutá aplikácia RESIQL Manager. Tá bola navrhnutá tak, aby bola jednoducho rozšriteľná o novú funkcionálnosť. Spolu s aplikáciou bola napísaná užívateľská príručka, ktorá poukazuje na možnosti vyvinutej aplikácie. Pre programátorov práca obsahuje technickú príručku so základnými informáciami potrebnými na zorientovanie sa v projektoch aplikácie. Do budúcnosti by mohol RESIQL Manager slúžiť ako administratívny nástroj na správu RESIQL databáz.

## Conclusions

Application RESIQL Manager was developed based on RESIQL database's features and supervisor's requirements. It was designed to be easily extensible with a new functionality. Together with the application user guide was written. Work contains technical guide with essential information for developers to work with the application projects. For the future RESIQL Manager could serve as an administration tool for RESIQL databases.



## A Obsah priloženého CD

Priložené CD obsahuje všetky dôležité materiály súvisiace s bakalárskou prácou.

### **bin/**

Obsahuje spustiteľný java súbory RESIQL Manager aplikácie s databázovým serverom a knižnicou ModuleLibrary. Priložená je aj java dokumentácia.

### **doc/**

Text práce v PDF formáte s priloženým zdrojovým textom v zip archíve na bezproblémové vygenerovanie PDF dokumentu.

### **src/**

Zdrojové kódy Eclipse IDE projektov ResiqlManager a ModuleLibrary.

### **literature/**

Materiály k jazyku RESIQL.

### **readme.txt**

Požiadavky a inštrukcie na spustenie aplikácie RESIQL Manager a lokálneho databázového servera.

## **Zoznam skratiek**

**DBMS** Database management system

**JRE** Java Runtime Enviroment

**RESIQL** Relational Similarity-based Query Language

## Literatúra

- [1] R. BELOHLAVEK, V. VYCHODIL. Query systems in similarity-based databases: logical foundations, expressive power, and completeness. In: *ACM Symposium on Applied Computing (SAC)*. 2010, str. 1648–1655.
- [2] R. BELOHLAVEK, V. VYCHODIL. Data tables with similarity relations: Functional dependencies, complete rules and non-redundant bases. In: *Database Systems for Advanced Applications*. In: *Lecture Notes in Computer Science*. 2006, str. 644–658.
- [3] CODD, E. F.. A relational model of data for large shared data banks. *Communications of the ACM*. 1983, č. 26, str. 64–69.
- [4] R. BELOHLAVEK, S. OPICHAL, V. VYCHODIL. Relational algebra for ranked tables with similarities: Properties and implementation. In: *Lecture Notes in Computer Science*. 2007, str. 140–151.
- [5] CHRISTOPHER J. DATE HUGH DARWEN. *Databases Types The Relational Model: The Third Manifesto*. 2006.
- [6] P. KRAJCA, V. VYCHODIL. *Basic Concepts of Relational Query Language for Similarity-Based Databases*. 2012.